opencorporates

Policy Paper

How OpenCorporates should handle company number problems

Overarching Principles for this paper

In approaching problems such as this OpenCorporates will follow the following principles:

- *Focus on end users.*
- *Transparency: wherever possible ensure any changes are done transparently.*
- *Clarity: ensure it is clear what decision has been taken, and why.*
- *Comprehensibility: ensure the decisions are explained fully, using examples where necessary (e.g. on the OpenCorporates register pages and the wiki).*

This document is expected to be revised over time, as we understand both the problem space, and possible solutions better. Significant changes will be denoted with a change in the version number (see footer).

Licence

This policy paper is copyright Chrinon Ltd (publishers of OpenCorporates) and published under the Creative Commons Attribution Licence 3.0. Please feel free to distribute.

About OpenCorporates

OpenCorporates is the largest open database of companies in the world with in excess of 90 million companies in 100+ jurisdictions. It regularly works with governments and intergovernmental organisations, and with NGOs and is dedicated to improving access to and understanding of company data in the public interest.

Version History

V1.0	*Oct 2015*	*Initial version*
V1.1	*Feb 2019*	*Updated approaches to be taken for companies where the public company number does not uniquely identify the company - use of governmental or GLEIF registration authority identifiers*
V1.2	*Feb 2019*	*Updated approaches for handling companies with more than one registration, including where relocation of a company causes a new registration*

Background

Well-designed company numbers – unique, persistent identifiers for companies issued by corporate registers – are an essential requirement for good company information *(we are using the term 'company numbers' to refer to unique identifiers issued by corporate registers – even those which are non-numeric*[1]*)*. Quite apart from ambiguities and normalisation issues, companies relatively frequently change their names, and legal names are even reused, meaning that such identifiers are the only way of categorically identifying legal entities. Unfortunately while most corporate registers issue such identifiers, OpenCorporates has discovered a number of registers where the identifiers are neither well-designed nor persistent nor unique.

This policy paper discusses the issues and outlines the routes OpenCorporates will take in handling them. In the spirit of transparency and openness that is at the heart of OpenCorporates we are making this paper public under a Creative Commons Licence, and welcome comments.

We have identified 3 distinct types of company number problems. We list these below, together with proposed solutions OpenCorporates will apply to address these in a transparent and open manner.

1. Registers changing company number system

In this situation, registers move from one identifier system to another. This is often done without announcement or explanation, and most seriously of all often without any mechanism or lookup table to allow a cross-walk from one identifier system to another.

Possible approaches:

a. Delete all the old records and add the new ones. This would have three problems – the old URLs would break; there would be no ability to search using the old identifier; the change would be invisible to user, lacking any audit trail. This approach has thus been <u>rejected</u>.

b. OpenCorporates should transparently map the old company identifiers to the new identifiers. There are issues with this approach, as it involves more work, and if there is no explicit or implicit mapping from the old identifiers to the new ones it means there are judgement calls in making the mapping. However, it means that we can both record and display the old company number (as "previous company number") and redirect from the old company URLs to the new ones. This provides both the transparency our users depend on, and allows the existing URLs/URIs to continue to function. In the case where OpenCorporates' mapping between old identifiers and new ones is in error, this too will be discoverable, and hence correctable. This is the approach OpenCorporates has decided to take.

Notes: Care must be taken before implementing that there is no overlap between the old identifiers and the new ones – if that is the case, the above approach will be problematic and should not be used.

---

[1] *The 'numbers' quite often include non-numeric characters and occasionally include punctuation (e.g. dashes), which may or may not be semantic*

2. Companies that have been issued more than one company number

There are three common reasons for this:

i. Change in company number (*see above*).

ii. Company numbers that contain 'state' or attributes of the legal entity, e.g. the location of current registration or type of entity (where this can change), or some other attribute. Examples include Indian CIN (which changes when a company changes company type or industry code, for example) or those jurisdictions such as Germany where the [identifier changes when the entity moves from one district court to another](#).

There are potentially two ways of dealing with identifiers that change depending on attributes:

   a. Create a new identifier from those parts that do not change, but combined uniquely and permanently identify the entity. We need to be certain that these parts do indeed uniquely and permanently identify the entity, and certainly a full investigation is required to do this. In cases where this is possible, this route would have many benefits. As with other situations, we would need to make this clear to users using native company number(s) and redirection, and with a public report. We should also ensure we redirect URLs based on the full identifiers to the compacted identifier. We should make clear what the current full identifier is (if possible), as well as the previous such identifiers

   b. Associate the previous and subsequent registrations, allowing users visibility of these "related registrations", by means of a "Subsequent Registration" mapping. This creates an explicit link between registrations, but does mean that we maintain multiple registrations (and thus OpenCorporates URLs) for a single legal entity. There is a precedent for this approach in the registration of out-of-jurisdiction branches (aka foreign corporations), which do not have separate legal personality but nevertheless have distinct registrations.

iii. simultaneous registrations for the same legal entity in multiple registers (excluding the case of branch registrations which are already handled by BranchRelationship objects). So far, the main example that we have identified is the situation in Germany and Switzerland, where multiple registrations could be maintained following a merger (Siemens AG has two registrations).

In this situation, if it is possible to associate the multiple registrations, then they should be cross-referenced by means of an "Alternate Registration" mapping, which makes explicit to users the relationship between registrations.

Arguably, the situation in France, where both Insée and Infogreffe hold information about the same legal entity is an analogous case, but for simplicity (and because they use the same identifier, the SIREN) we will for the moment maintain the position of merging the data before import, and treating as a single register (see this [blog post](#)).

3. Companies where the public company number does not uniquely identify the company.

We have seen this where:

  i.   Company numbers have been reused by the register so that the same identifier is used by multiple entities.

  ii. Company numbers are unique only to entity type. In some cases there are obvious and visible codes used by registers to denote the entity type; in others there is no such obvious route.

  iii. Company numbers are unique only to the sub-registration body. This is the case in Germany, where the identifier is a mixture of the district court (Amstgericht) and a combination of the register type and a numeric identifier.

  iv. There is more than one register per 'jurisdiction' (e.g. California corporations and LLCs, Australian companies and Australian Indigeneous Corporations).

Proposed approaches:

  i.   Where company numbers have been reused, and there is no way of distinguishing between these by type of company, for example, we will either need to use an alternative identifier (e.g. an internal database ID that is exposed in the URL or page parameters) or defer loading the jurisdiction till an alternative solution can be found. While using such an alternative identifiers is problematic, as they are liable to change without notification, it is in general more important to users to make the company information available to users (there is also the option to handle such changes via the change procedure outlined in 1b above).
  We should communicate this clearly to users by showing the basic number as 'native company number', listing the issues on the register page, and writing a report on the OpenCorporates wiki explaining what has been done, with examples. This fact that these identifiers are different from the normal approach should be further communicated to users by prefixing the internal ID with EXTUID_ in the URL/company number fields (e.g. https://opencorporates.com/companies/je/EXTUID_63953).

  The native company numbers should also be made discoverable by indexing them so that searches for them will be successful. Finally, if there were companies that had OpenCorporates entries prior to the problem being discovered, redirects should be put in place so that the old URLs still work for users.

  ii. Where company numbers are unique only to one entity type and there are obvious entity prefixes, we can use these in the OpenCorporates identifier, e.g. RC_12345. However we must be certain that the entity cannot change company type (e.g. from private to public company) and thus prefix before taking this approach. In addition, we must make this to clear to users by both displaying the basic (possibly duplicated) identifier as 'native company number', and make this discoverable by indexing it so that searching for it will return the companies with this number.

  Where there are no obvious entity prefixes, or we believe that there are other problems with

the data, we should use the EXTUID approach detailed in *(i)* above, or hold off importing the jurisdiction until we have an approach that meets the standards above (user-focused, transparent, clear, comprehensible).

iii. Where there is more than one register per 'jurisdiction' we should distinguish between two different situations:

  a. Registers that are distinct for largely technical reasons, e.g. California, which stores the domestic LLCs in a different register from corporations and out-of-jurisdiction LLCs. In some cases these have identifier systems which play well together (i.e. non-overlapping), and if this is the case we may well be able to present this to the user without any issues – that is, if the separation is just technical but neither legal nor logical.

  If there is a risk of overlapping numbers, or if the numbering system of one of the registers is unknown, we can use a prefix for the registers to distinguish (or namespace) the identifiers. The prefix should be either formal, well described ones issued by the government itself[2], or if these do not exist, the identifiers issue by the GLEIF in their [Registration Authorities List](#). e.g. RA000693_123 to identify the legal entity with local number 123 in the Belize IBC Register (RA000693 is the GELIF RA code for the Belize IBC Register. If there is no governmental, or GLEIF identifier for the register, we will need to defer importing the register until we have refined the policy and potentially the tech to handle such situations.

  b. Registers that relate to different jurisdictions (for example [Australian Indigenous Corporations](#) are arguably a distinct jurisdiction to the ASIC company register). For this case, we should in general recognize it as being distinct in the way, for example that US states or Canadian provinces are distinct. Where there is no ISO 3166-2 code for the jurisdiction, we could potentially use the [GLEIF Registration Authorities](#) identifier.

---

[2] This is the approach taken by the EU for the EUID: see EU (2015/884 Annex s.8)